TUTORIAL TEST SUIT SPACEWIRE

LATIN AMERICAN GROUP INTEGRATED CIRCUITS

GLADIC

Felipe Fernandes da Costa
Modeling Architect

WHAT THIS PROJECT WISH HELP YOU

- **Don't make you lose your head when you try run this**
- **Explain in basic way how to use this**
- **Make tests with a SystemC model Spacewire**
- **A test suit using graphical interface**

OS type: Linux based rpm distro

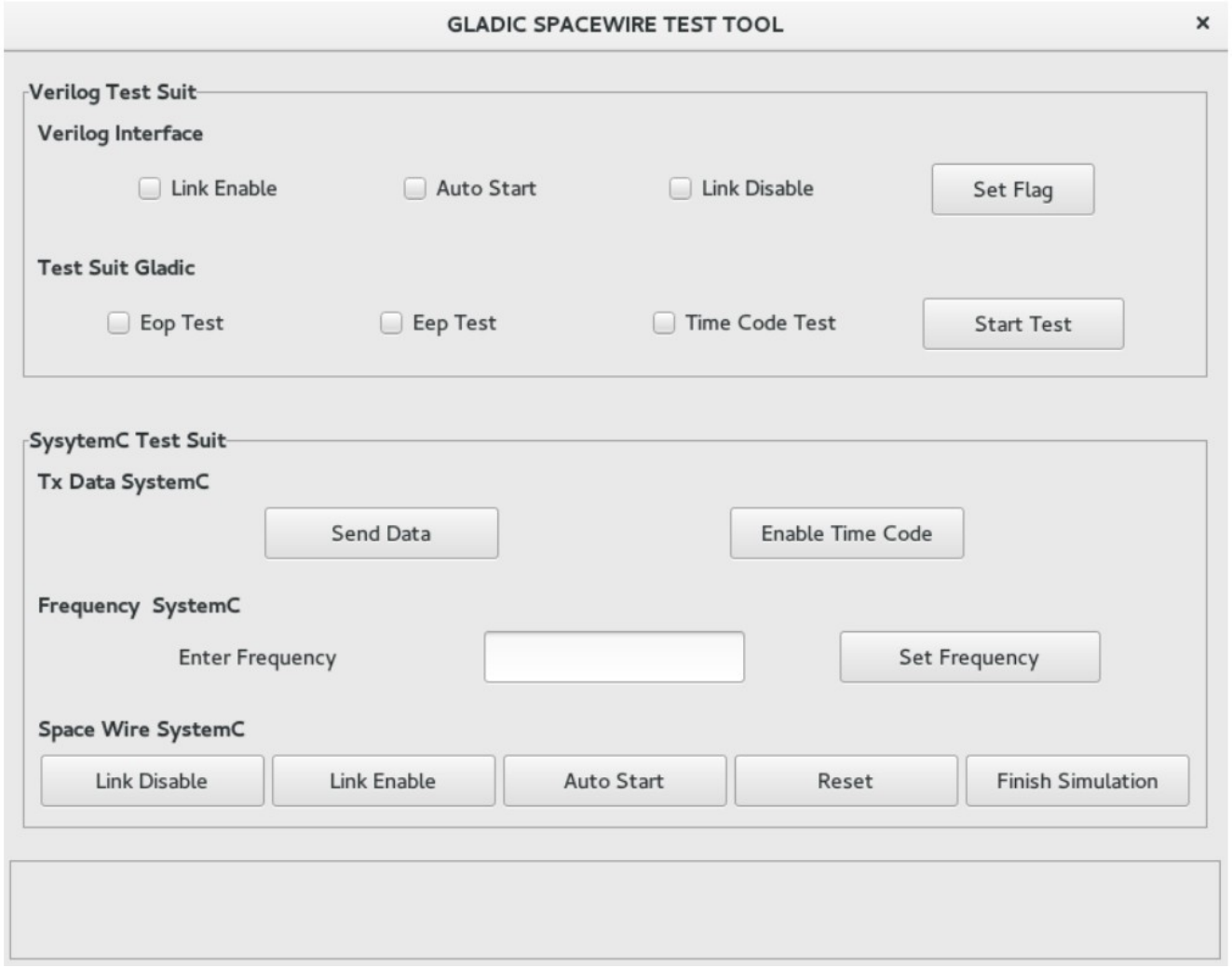
Graphical Interface: GTK/GTKMM (minimal 2.16)

Compiler: g++/iverilog

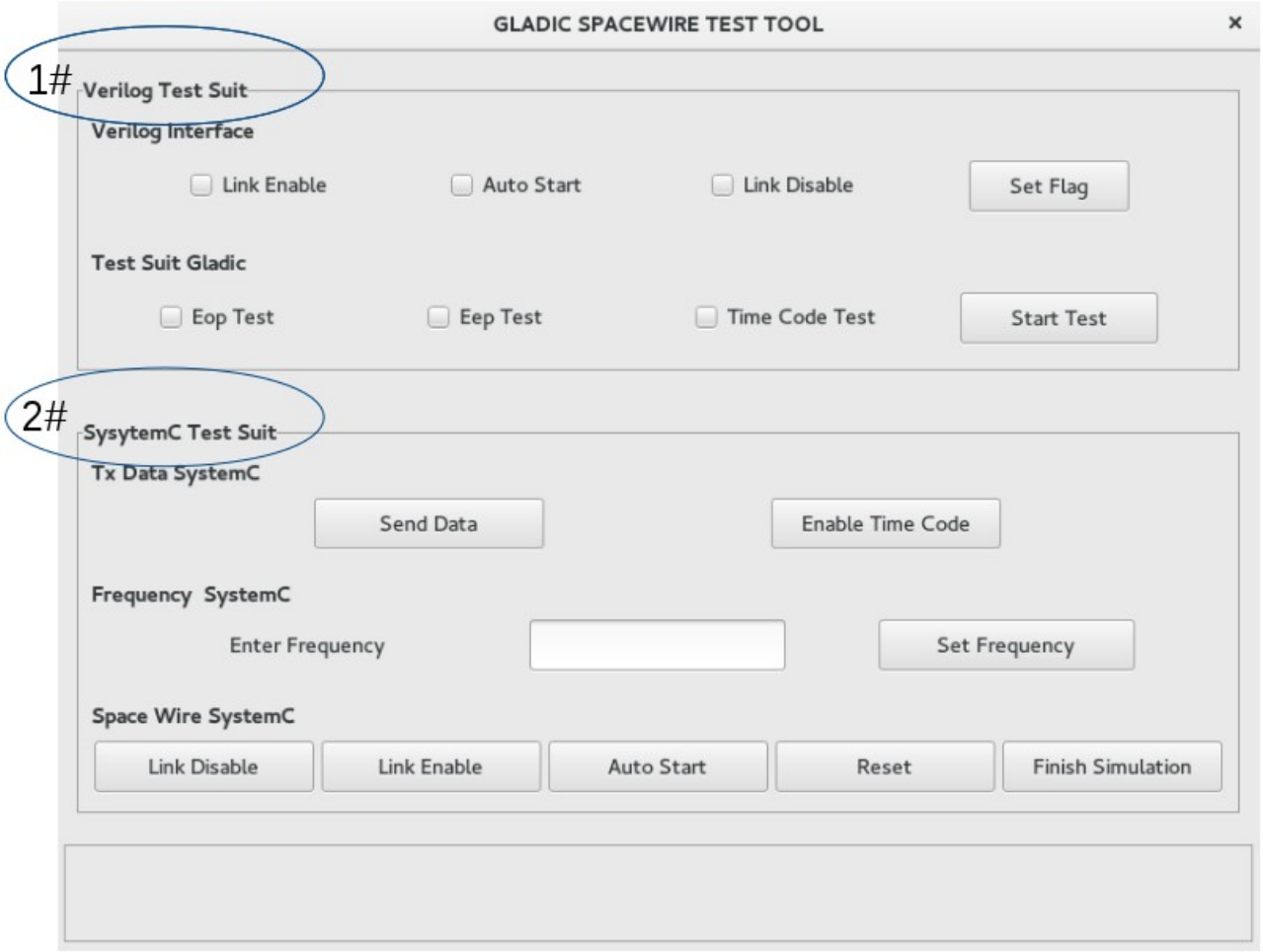
API: boost,systemC,gtkm,gtkmm,glade

Obs: Need take a look on the flags during compilation to make sure are you compiling and link correctly with shared library. SystemC need to be compiled in separate and need to exported in order to code see systemC shared objects like icarus verilog vpi source code.

GRAPHICAL INTERFACE



TEST SUIT SYSTEMC / VERILOG

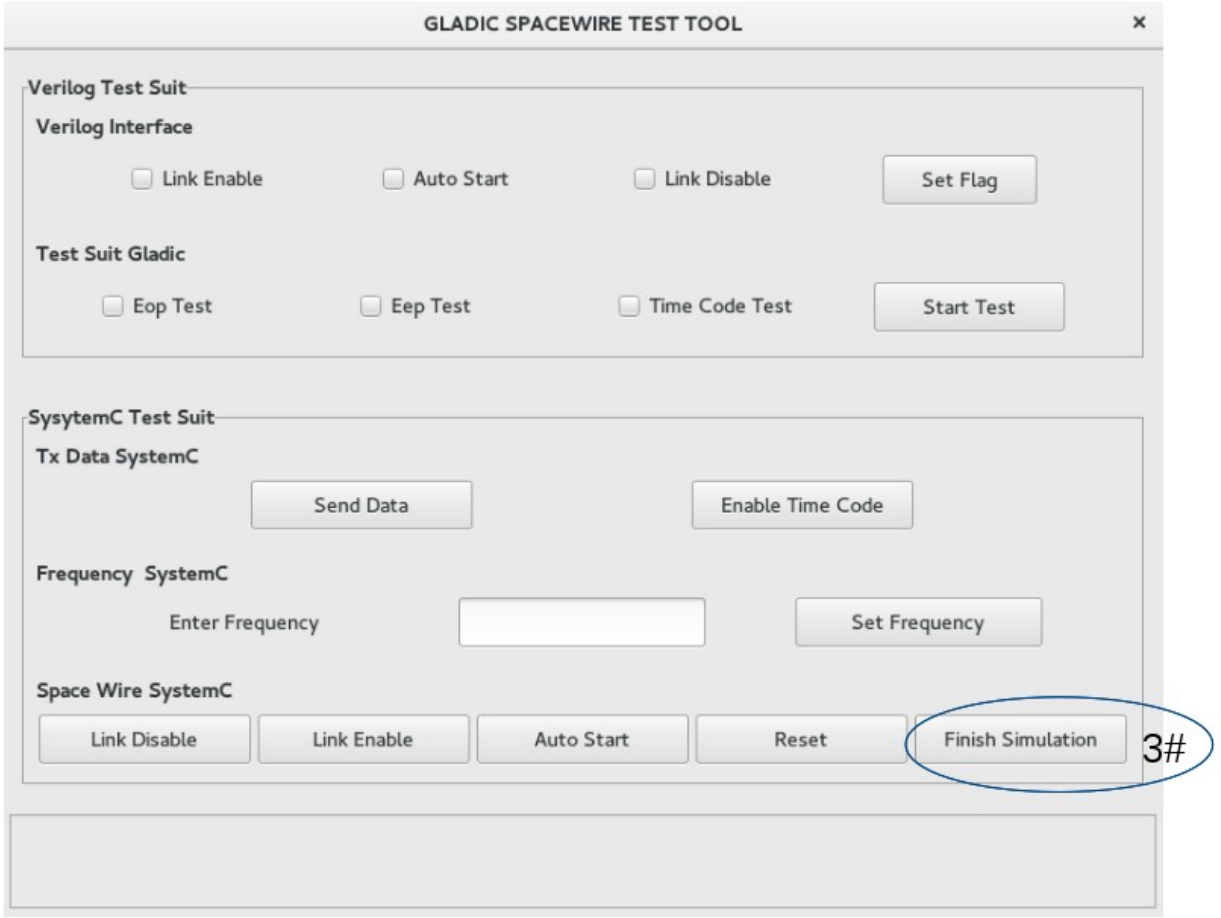


TEST SUIT SYSTEMC / VERILOG

1# This part consist in set pins of verilog side environment. In another words you have control by buttons. This aim validation to make specific case in order to check errors or validation where you can proof in time simulation the correct work of verilog/systemverilog hardware develop.

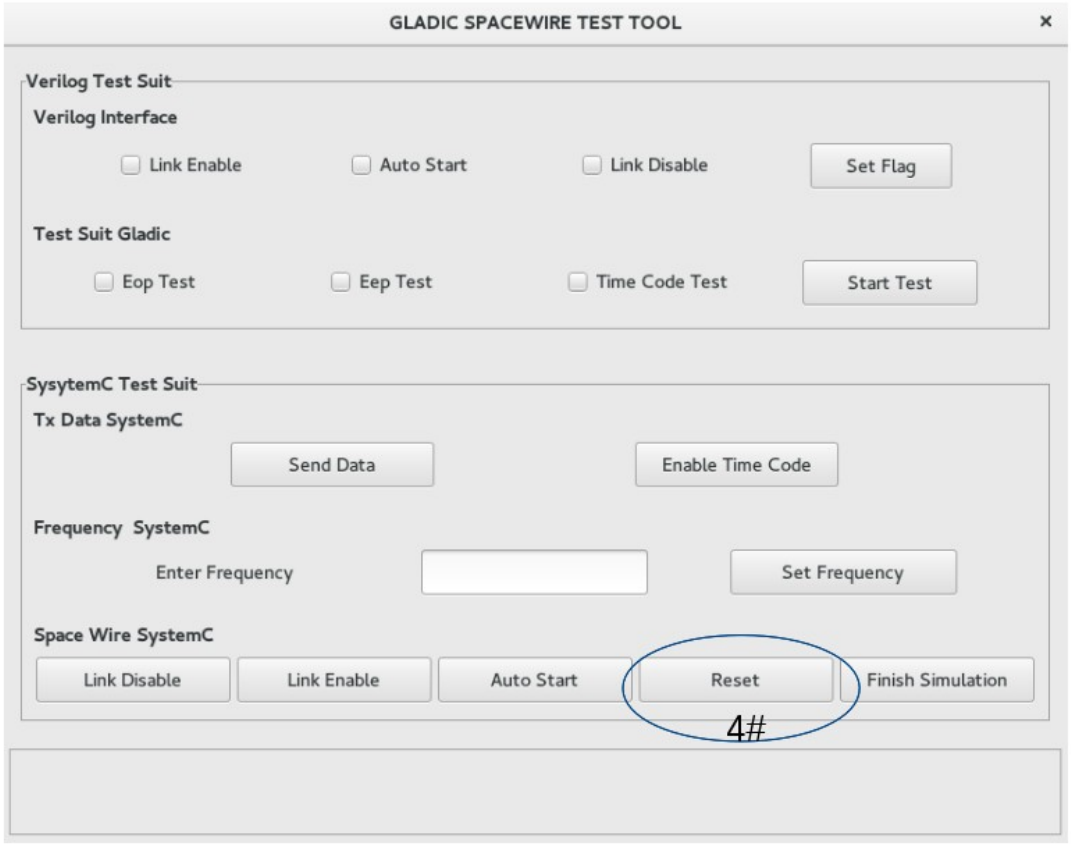
2# Now the part of environment. This contain the control over all simulation. You can change frequency rate from , reset entire system, generate data, time codes.

TEST SUIT FINISH SIMULATION



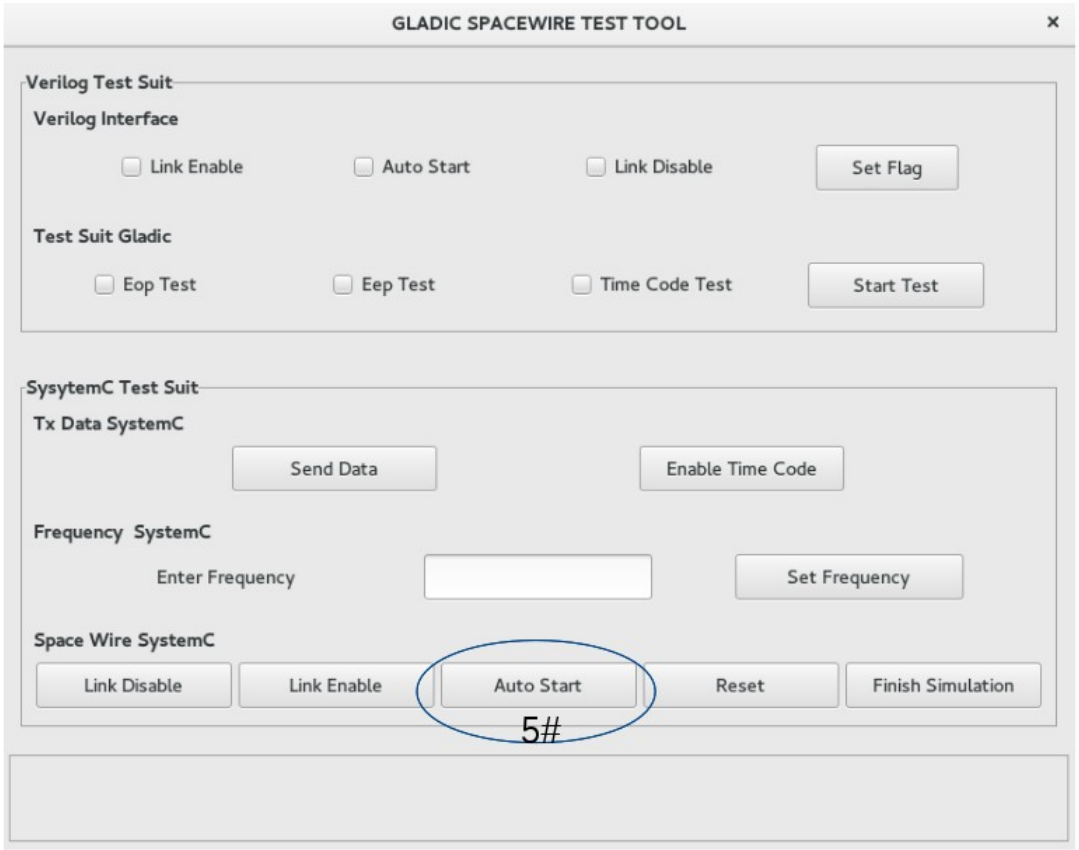
3# This button finish simulation of environment systemC and verilog setting a variable on verilog to finish simulation and quit from icarus verilog simulator.

TEST SUIT RESET



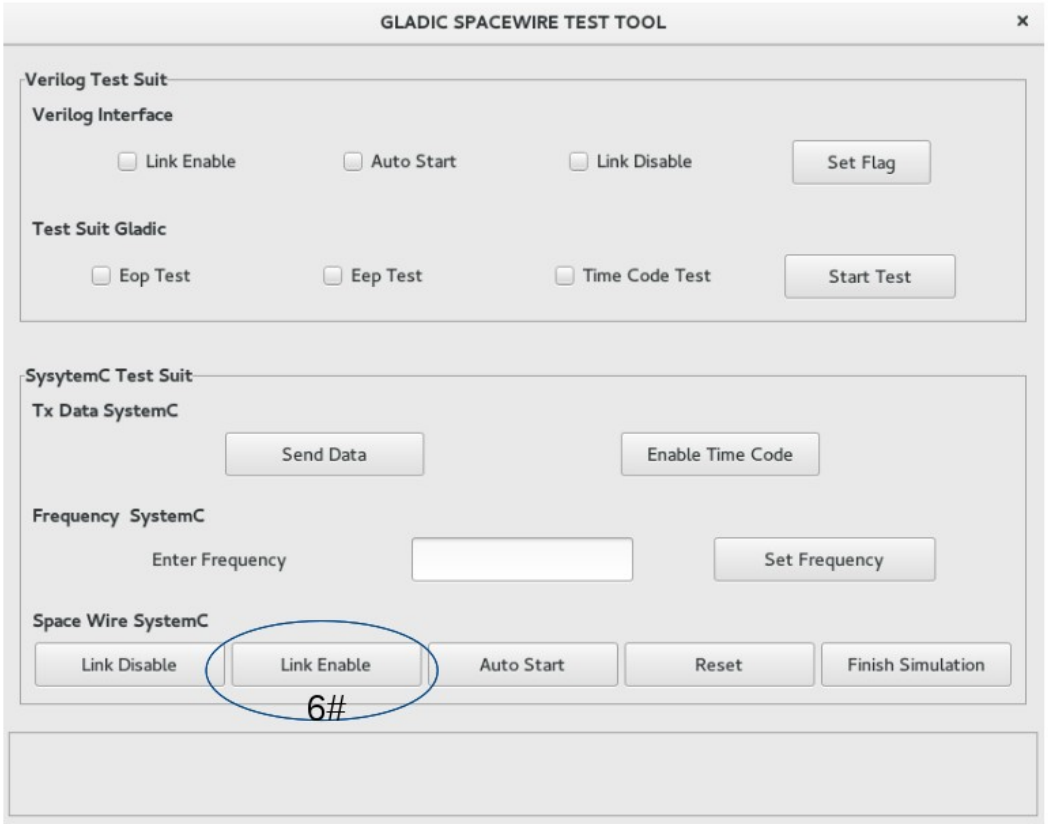
4# This button reset all the environment. When environment is executed by the first time you need reset both model and rtl SpaceWires. Not resetting you can expecting wrong behavior and crash execution.

TEST SUIT SYSTEMC AUTOSTART



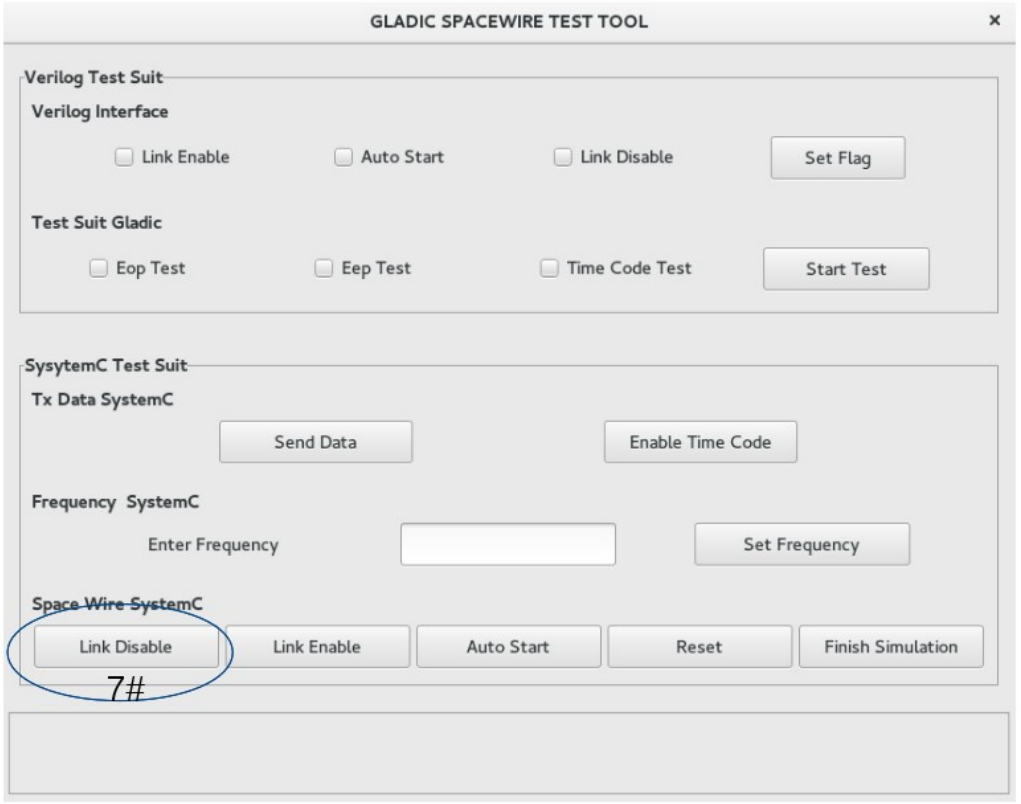
5# This button is present on specification on the SystemC model. In the way to we use it like a spacewire AutoStart wait another Spacewire to start communicate to start send NULL's" ECSS-E-ST-50-12C(31July2008)". Before first reset Auto Start is not enabled.

TEST SUIT SYSTEMC LINKENABLE



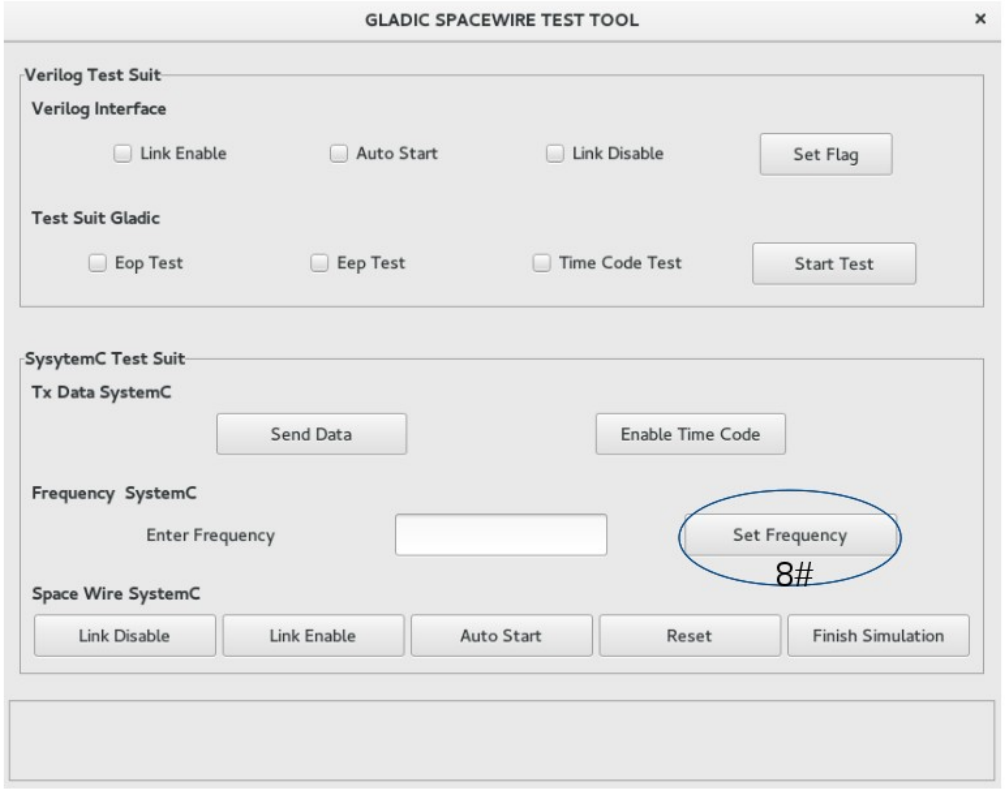
6# This button is present on specfication on the SystemC model. In the way to we use it like a spacewire Linkenable initialize the line sending NULLs to another Spacewire to start communicate and send NULL's “ ECSS-E-ST-50-12C(31July2008)”. Before first reset LinkEnable is not enabled.

TEST SUIT LINKDISABLE



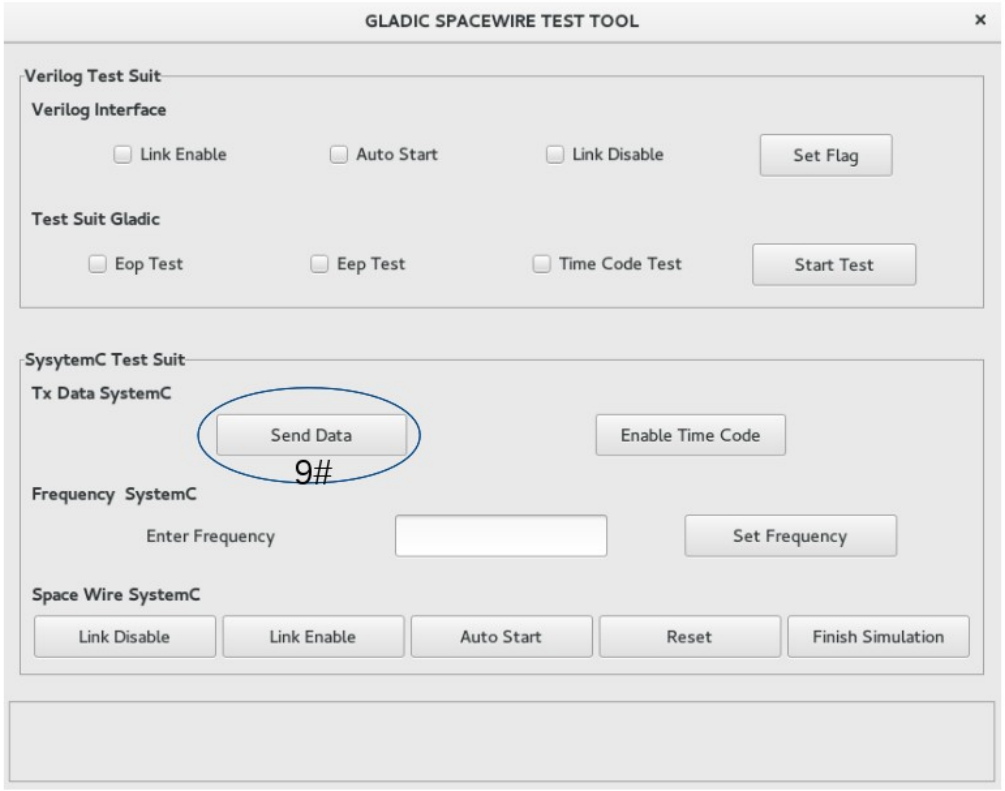
7# This button is present on specification on the SystemC model.This button disable spacewire according with definitions ECSS-E-ST-50-12C(31July2008). So when are you trying connect or on state run and this button is set the model will disconnect Before first reset LinkDisable is not enabled.

TEST SUIT FREQUENCY TX MODEL



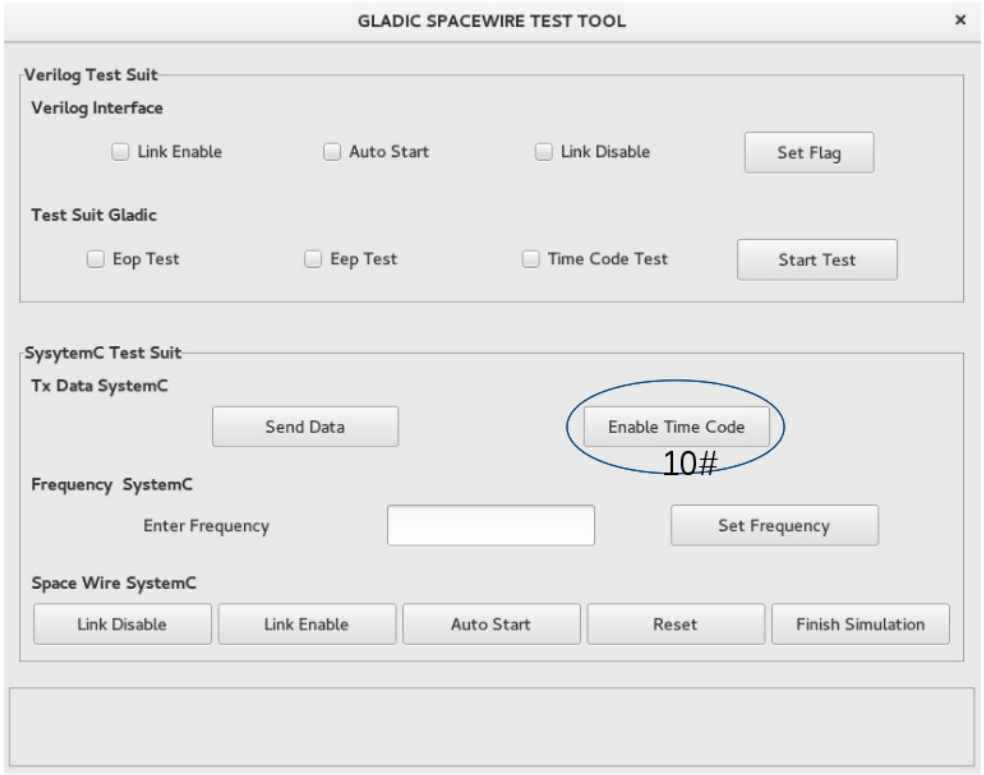
8# This give the chance to set frequency on TX spacewire in order to test. The valid frequencies are : 2 – 10 – 20 – 50 – 100 – 150 – 200 – 201 – 250 – 280. After the first reset TX model start at clock operation 10 MHz “May be changed during connection”. Can change during data transfers/timecodes to check any problem in change frequency on RX verilog.

TEST SUIT DATA TX SEND



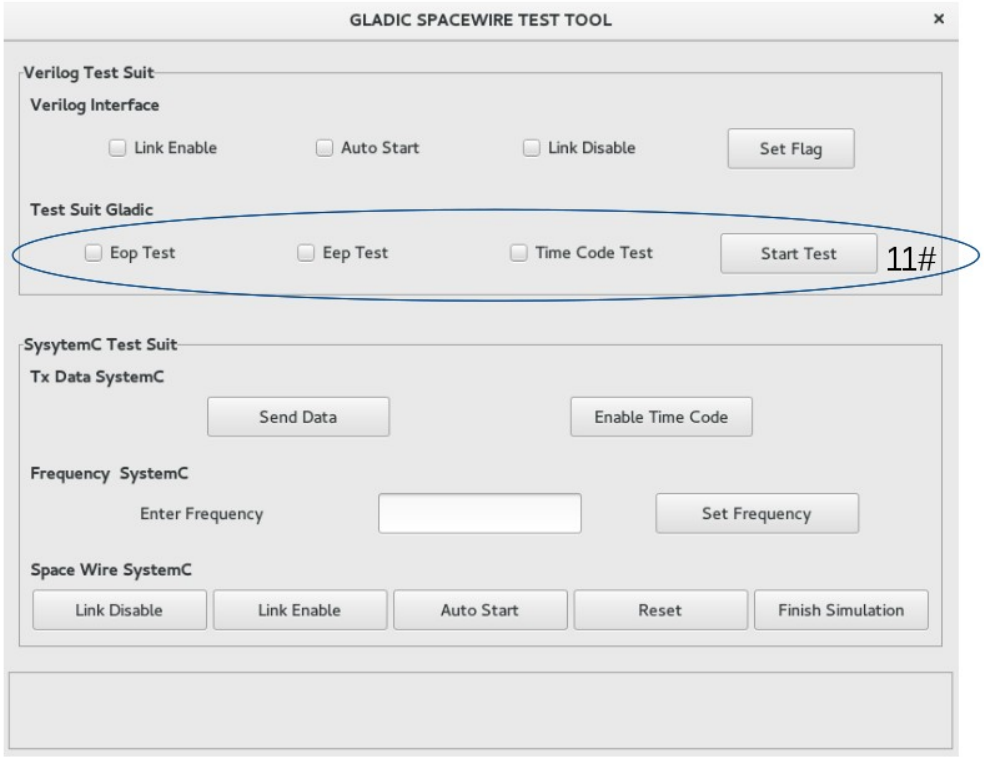
9# You can send data from model to verilog. This generate a htm log. To get a better understand this button can be used only after you send data from verilog to systemc. It crash if you try without generate to verilog first.

TEST SUIT TIMECODE TX SEND



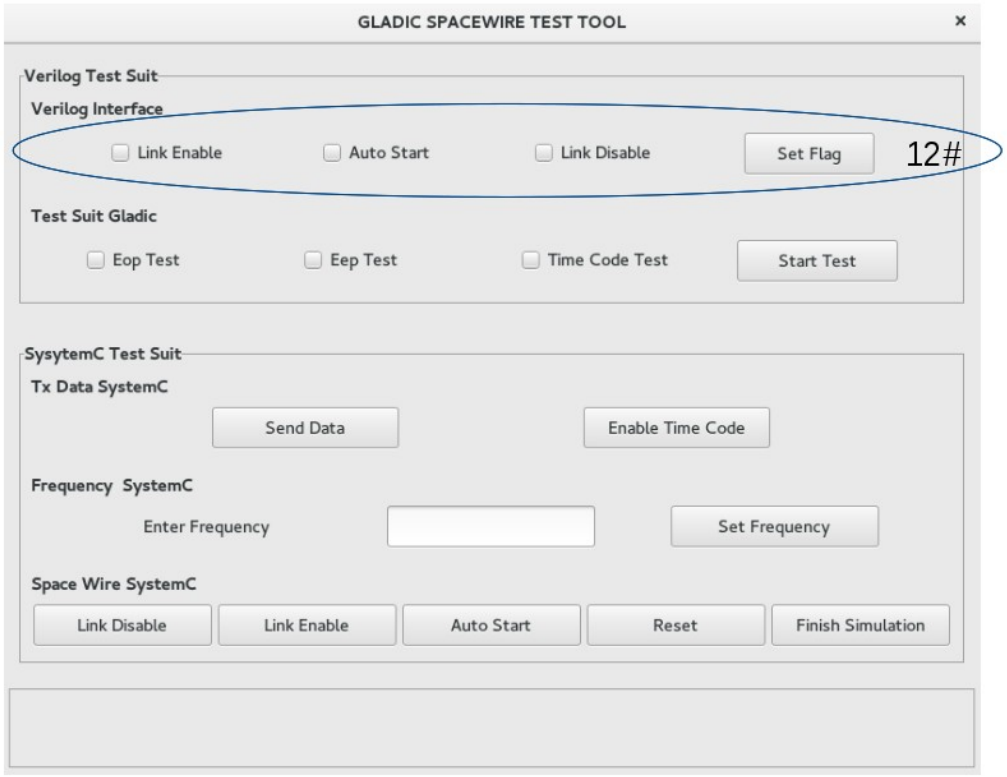
10# Time code is generated automatic when you hit the button. After and before reset timecode is disabled.

TEST SUIT N-CHAR TX SEND TEST



11# The Nchar test consist on build a package EOP - DATA - EOP ; EOP – DATA - EEP – EOP ; EOP – DATA/EEP/TIMECODE – EOP. EOP have high priority against EEP; time code can be executed in same time EOP/EEP tests.

TEST SUIT VERILOG PINS INTERFACE



12# This obey in the same way like described using SystemC." ECSS-E-ST-50-12C(31July2008)".

TEST SUIT OBSERVATIONS

- Still under development “issues about errors and data time between spacewires ”
- We hope have some feedback”Help us and we help you”
- Donations are welcome !!!!! “nobody can survive eating systemc/systemverilog everyday” :-D
- Design is reluctant a bit cause bad code, but I hope he post soon as possible his solution “systemverilog”
- The environment is running on 500 MHz clock to be possible simulate SystemC/SysTemVerilog over 200 MHz
- There is two files on html where contain data comparison and what arrive from the line both sides
- If you have a question about configuration of environment enter in contact by github → issues